Attempting to Correct the USA Energy-Dependent Instrumental Effect in the Time Domain

Kent S. Wood and Paul S. Ray

Naval Research Laboratory

1. Introduction

The Energy-Dependent Instrumental Effect (EDIE), also known as the "400 Hz bump" has been described in detail elsewhere, so we won't explain it here except for a brief summary of the relevant details. The current working model for this effect is that long tails on the pulses from detector events can cause the energy of photons which come later to have their energy shifted and place them in the wrong energy bin. This is a kind of pileup effect since it is due to the voltage pulses from seperate events getting summed. Because the photons getting their energies bumped come at preferred times (*i.e.* within a certain time window) after other photons, this causes a modification to the autocorrelation function of the photons in a particular bin. What should be white noise Poisson data becomes distorted and this can be seen in the autocorrelation function or power spectrum of the resulting data. The effect thus depends on the pulse shape from a photon of a given energy and the energy spectrum of the photons hitting the detector. Large gradients in the number of photons from energy bin to energy bin make the effect much more dramatic.

Here, we describe the beginnings of a method for correcting this effect in the time domain. This work was motivated by the fact that what we are typically trying to measure from the USA data is a power spectrum, which contains only about half the information of the original time series (since the phases are not retained). Also, note that the autocorrelation function contains exactly the same information as the power spectrum since they are related simply by a Fourier transform (via the Wiener-Khinchin Theorem). A correction method thus does not need to reconstruct the original time series perfectly, but rather only has to construct a time series which has the same autocorrelation function as the original time series.

2. Method

For this and the following discussion, we are going to treat the effect in an idealized manner, as if there were only two energy bins and the pileup could only knock photons down from the upper channel to the lower channel. For actual application to USA data the pileup probability distribution and the fact that there are channels above and below the target channel will have to be considered. This should be a straightforward generalization.

Consider our two channel time series. The upper channel (call it channel U) is populated with a bunch of events which had the correct energy to fall in that channel. It is also missing some

(unknown) quantity of events which, due to the pileup problem, were mis-identified as being in the lower channel (call it channel L). Correspondingly, channel L contains a bunch of events which were correctly assigned to that channel plus the set of events which should have been in channel U but got bumped down. Unfortunately, it is not possible to tell which of the channel L events should have been in channel U to correct them. We also need a set T for the total of all the events in all of the valid channels. In this case T is just all of the events in L and U.

The idea is that looking through the events in channel L, one can identify the events which *could possibly* have come from channel U because they will be within R μs of an event in T. Here R is the full range of the effect, *i.e.* the longest interval between photons where the second photon can get bumped out of its correct channel. Note that photons can get bumped by the tails of pulses from vetoed particles whose times are not recorded. These can be ignored because the distribution of these events is completely white and uncorrelated with the X-ray events so does not affect the autocorrelation function of the X-ray events. For each photon in L which could have come from U, we put a fractional photon back into U at that location in time. See Figure 1 for a grapical representation of this process. The value of the fractional photon is based on the probability that it actually came from U and the number of photons in L which are likely to be mistakenly bumped back up to U. This can be calculated from the rates in U and L.

The hope is that the fractional photons contribute to the autocorrelation function with the correct time dependence to make up for the photons which were dropped, thus restoring the undistorted power spectrum of the original U time series.

3. Simulation

To test the concept, I wrote a quick test program in IDL. This is meant to be a toy model to test the concept, not an accurate description of the actual effect such as the Monte Carlo that Elliott has done.

The simulation begins by creating the U and L channels as perfect Poisson data each with their own setable rates. The simulation works with 32 μs bins for all time series. Deadtime is simulated by limiting each bin to 1 photon. (This is not quite accurate since this is technically a deadtime of 0–32 μs depending on where the first photon in the bin lands.)

Next, a new version of channel U is created where photons which are close to another photon in T have a certain probability of being put into L instead of U. This new version of channel U and L are considered as the input to the correction algorithm. The correction algorithm looks through the new channel L and finds all photons that could have come from U. This number will be more than the actual number of photons that came from channel U. These photons are put into a "fixed" version of channel U as fractional photons.

The normalization of the fractional photons is crucial to the algorithm. I began by choosing a

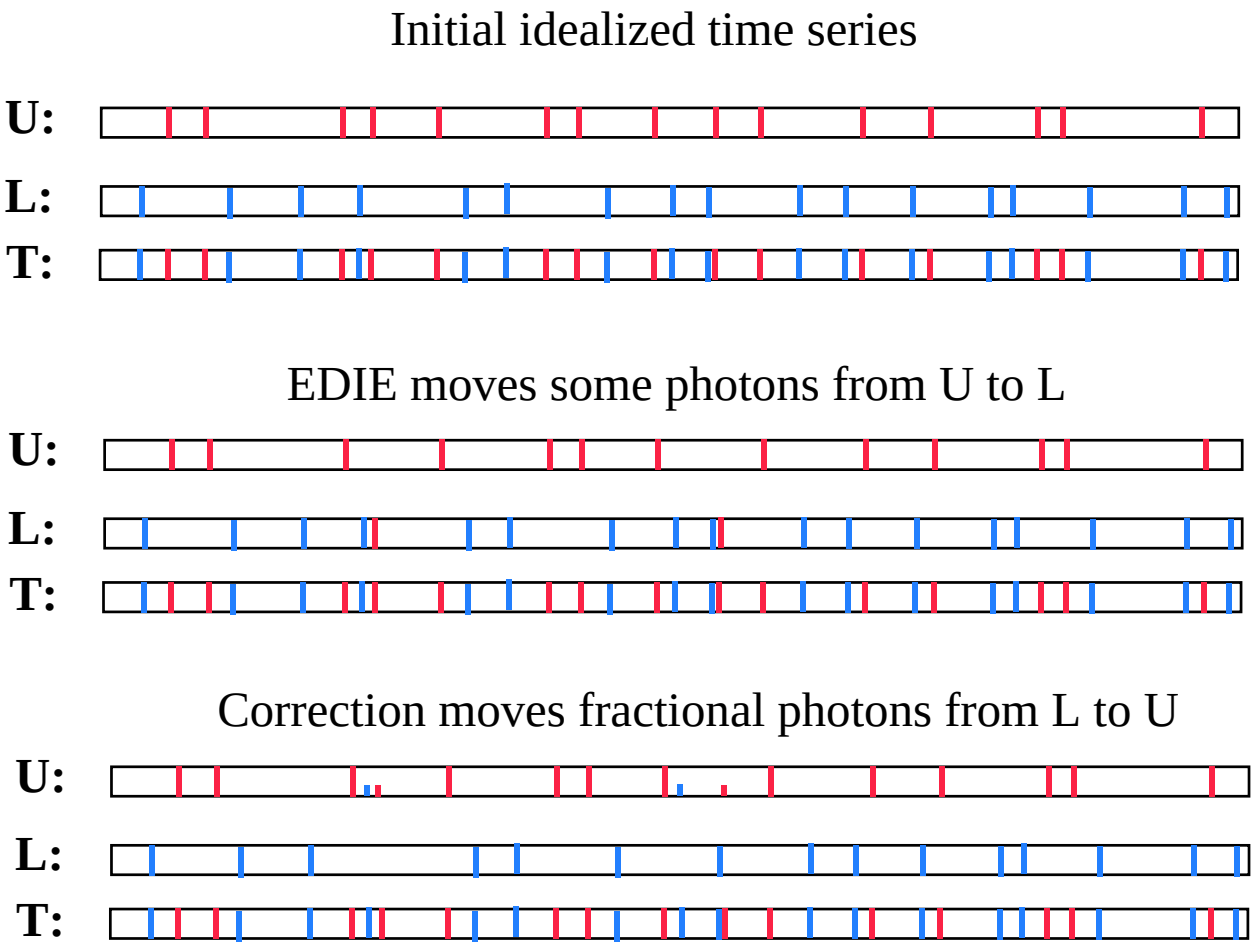

Fig. 1.— Diagrams showing fixing procedure. Note that the colors are to help track which bin the photons came from. In real life, events aren't painted with the color of the bin they should have landed in! The top panel shows the two channel time series that would have been recorded if the EDIE did not exist. In the second frame some of the photons from U (which are colored red) have been bumped down to L by the EDIE. This is what the raw USA data would look like and the power spectrum of U would be corrupted. The bottom frame shows fractional photons restored to U by the correction procedure. Note that both red and blue photons can get bumped up since it is not known which photons in L came from U. The Fourier power spectrum of this fixed U should be flat Poisson noise.

value which should keep a constant power by Parsevals Theorem[1], that is I used

$$f = (N_{\rm drop}/N_{\rm fix})^{1/2} \tag{1}$$

where $N_{\rm drop}$ is the number of photons from U that got dropped into L and $N_{\rm fix}$ is the number of fractional photons that got pushed up to U from L in the fixing process. This should ensure that the fixed version of U should have the same total Fourier power as the initial, uncorrupted U.

A concrete example is shown in Figures 2 and 3. For this simulation, the U and L time series were 4 Msamples with a bin size of 32 μs (corresponding to an observation of 134 s). The rate in the U channel was 500 counts/s and the rate in the L channel was 5000 counts/s. The corrupting process dropped 2450 out of 66503 events from U to L. Because of the high rate in the L channel, 470,102 fractional events got put back up into U in the fixing process (out of a total of 621,971 events in L). The fix factor calculated from Parseval's Theorem was 0.0722. This case was chosen as a test of a case where many more events get corrected back up to channel U than got dropped from U in the first place.

With the code, it is instructive to play around changing the ratio of the counts in U and L. I found that in many cases the correction produced by equation (1) overcorrected and turned the power deficit into an excess and had to be turned town a little bit. However, the algorithm does appear to work over a large range of parameters.

4. Conclusions

This method has shown encouraging ability to recover a flat, white-noise power spectrum from a corrupted time series. There are several ways to continue from this point.

- Try a non-white initial power spectrum to see if that can be recovered.
- Improve calculation of fractional photon amplitude, using only quantities measurable from the actual data.
- Add a more realistic treatment of the pileup pulse shape and probability of being bumped. These probably can come from Elliott's simulations.
- Add a more realistic treatment of deadtime
- Add channels above and below the target channel or range of channels.
- Try on real data!

[1]Parseval's Theorem states that the sum of squares in the time domain equals the sum of the power (squared Fourier amplitude) in the Fourier domain

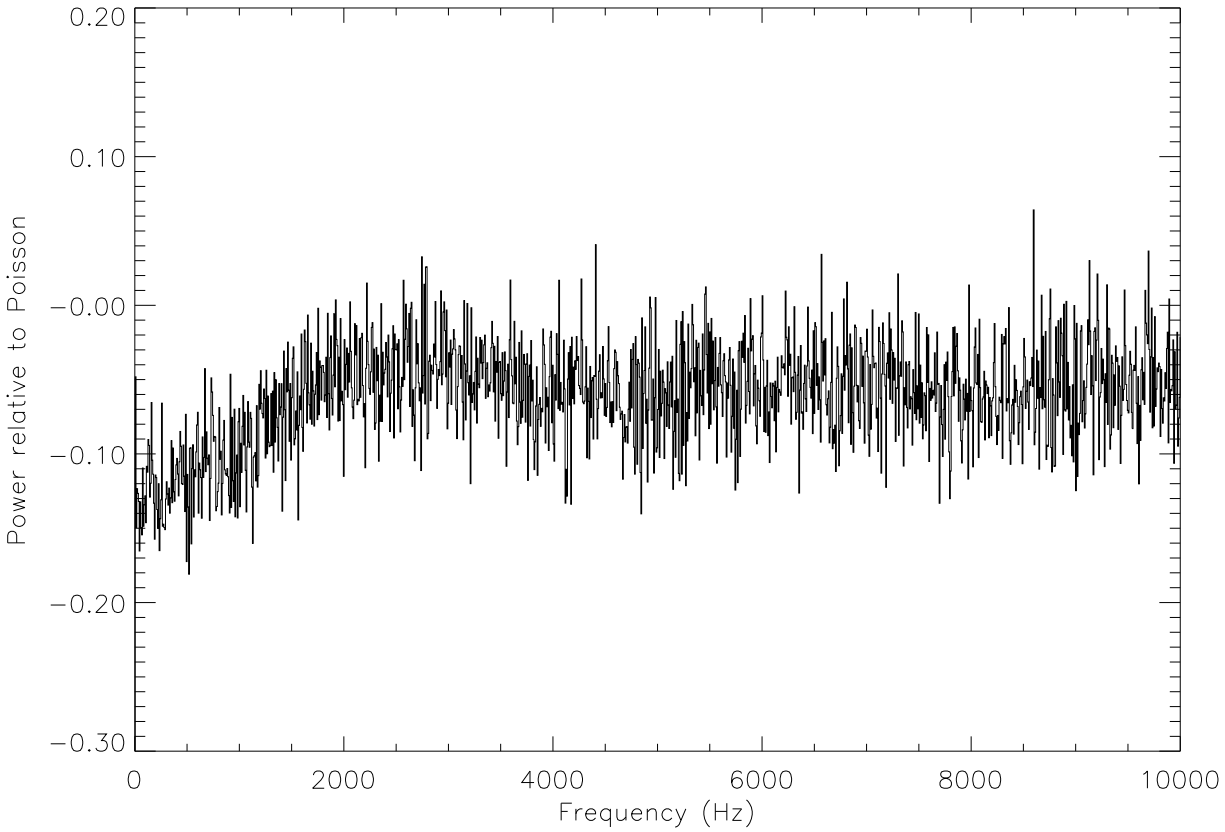

Fig. 2.— Spectrum corrupted by the EDIE as described in the text.

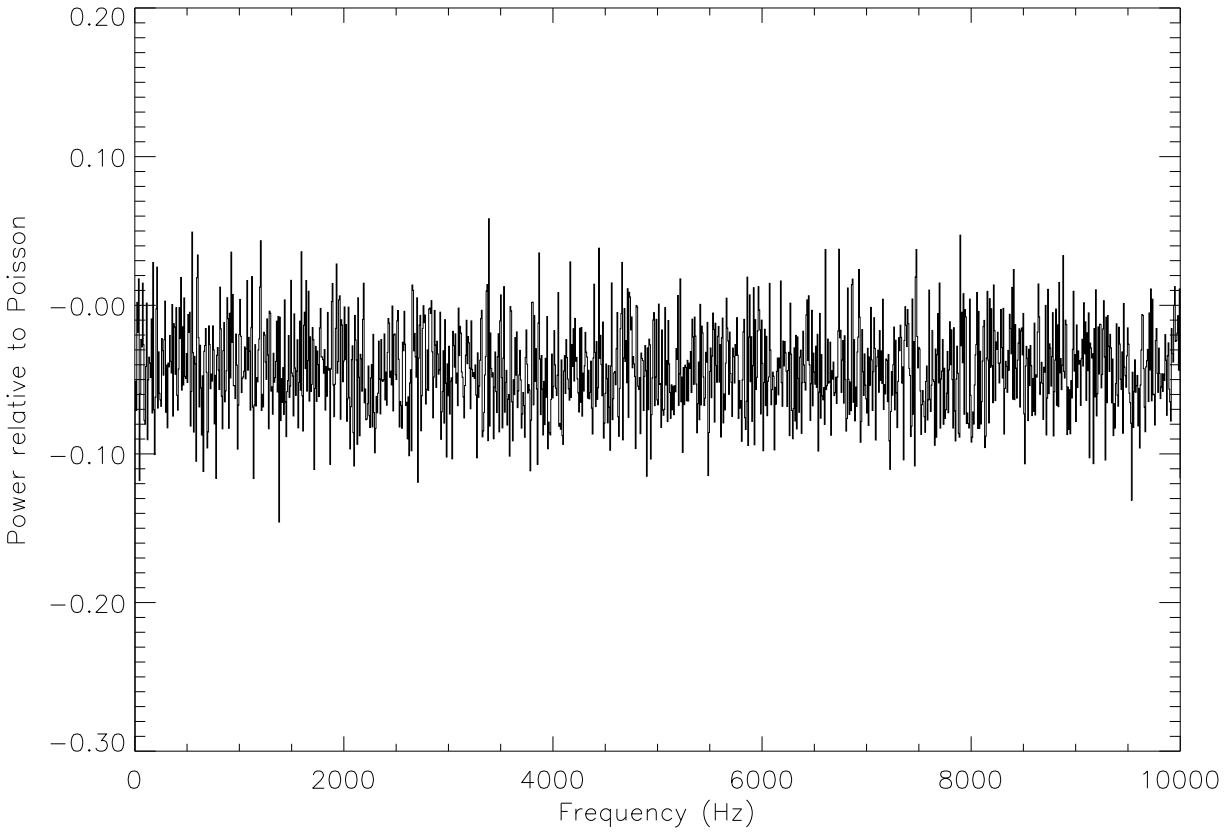

Fig. 3.— Spectrum after fixing by the procedure described in the text.

A. IDL Code Listing

```
;
; Try replicating the 400Hz bump effect in USA data and correcting for
; it using Kent's recipe.
;
; I am a novice IDL programmer, so forgive the inelegance.
;
; Paul Ray
; Created 2001 March 8
;

; set various things
dt = 32.0E-6
nbins = 4096L
span = nbins*dt
ntotal = nbins*1024L

; Set the range of the effect and its probability
efbins = 8
efprob = 0.3

ratehigh = 500.0
ratelow  = 5000.0

; Create two time series
lchigh = RANDOMN(seed,ntotal,POISSON=ratehigh*dt)
lclow = RANDOMN(seed,ntotal,POISSON=ratelow*dt)
lctotal = lchigh + lclow

; Simulate deadtime of 1 bins (dt width) by limiting bins to 1 photon
; This is not accurate since the deadtime then depends on where in the bin
; the first photon landed.
lchigh=(lchigh<1.0)

; Start with blank version of lchigh
lchighnew = fltarr(n_elements(lchigh))
; Copy lclow since all those photons will be there no matter what
lclownew = lclow

; Now for each photon in lchigh drop some into lclownew if
```

```
; they are close to a previous photon in lctotal with some probability
index = where (lchigh ge 1)
npass = 0L
ndrop = 0L
for i=0L, n_elements(index)-1 do begin
  ind = index[i]
  if ((total(lchigh[(ind-efbins)>0:ind-1]) gt 0) and (randomu(seed) lt efprob)) then begin
    lclownew[ind] = lclownew[ind] + lchigh[ind]
    ndrop = ndrop + 1L
  end else begin
    lchighnew[ind] = lchigh[ind]
    npass = npass + 1L
  end
endfor

print,"Passed ",npass," and dropped ",ndrop," events."

; Now try to fix lchighnew by looking through lclownew and
; pushing some photons back up into the high channel
lchighfix = lchighnew
index = where (lclownew ge 1)
nfix = 0L
nignore = 0L
for i=0L, n_elements(index)-1 do begin
  ind = index[i]
  if (total(lctotal[(ind-efbins)>0:ind-1]) gt 0) then begin
    nfix = nfix + 1L
  end else begin
    nignore = nignore + 1L
  end
endfor
fixfactor = sqrt(float(ndrop)/float(nfix))
print,"Fixfactor = ",fixfactor
for i=0L, n_elements(index)-1 do begin
  ind = index[i]
  if (total(lctotal[(ind-efbins)>0:ind-1]) gt 0) then begin
; The sum in the next line is the crux of the problem
    lchighfix[ind] = lchighfix[ind] + fixfactor;
  end
endfor
```

```
; Chop bins down to 1 photon maximum
lchighfix = (lchighfix < 1.0)
print,"Fixed ",nfix," and ignored ",nignore," events in lclownew."

; xx holds the time series that we are going to analyze by making power spectra
; lchighnew is the time series with the corruption
; lchighfix is the same with the correction applied in the time domain
;xx = lchighnew
xx = lchighfix

print,"Parseval's Theorem Before ",total(lchigh*lchigh)," After ",$
total(lchighnew*lchighnew)," Fixed ",total(lchighfix*lchighfix)

; Loop through the array and take the power spectrum
psum = fltarr(1+(nbins+1)/2)
pnum = 0
for i=0L, ntotal/nbins-1 do begin

   x = xx(i*nbins:i*nbins+nbins-1)
   fx = fft(x, -1)
   p = float(fx*conj(fx))
   p = p(0:(nbins+1)/2) ; discard negative frequencies
   p(0) = 0 ; discard DC component
   np = n_elements(p)
;   pmean = total(p)/np
; Normalize to TRUE Poisson level from original data
   pmean = span*ratehigh/nbins^2
   p2 = p/pmean ; mean noise level is unity
   df = 1.0D/span
   freq = df*dindgen(np)
   psum = psum + p2
   pnum = pnum + 1

endfor

; Now plot the power spectrum
  plot, freq, (psum/pnum)-1.0, psym=10,$
        xtitle='Frequency (Hz)', ytitle='Power relative to Poisson'

end
```